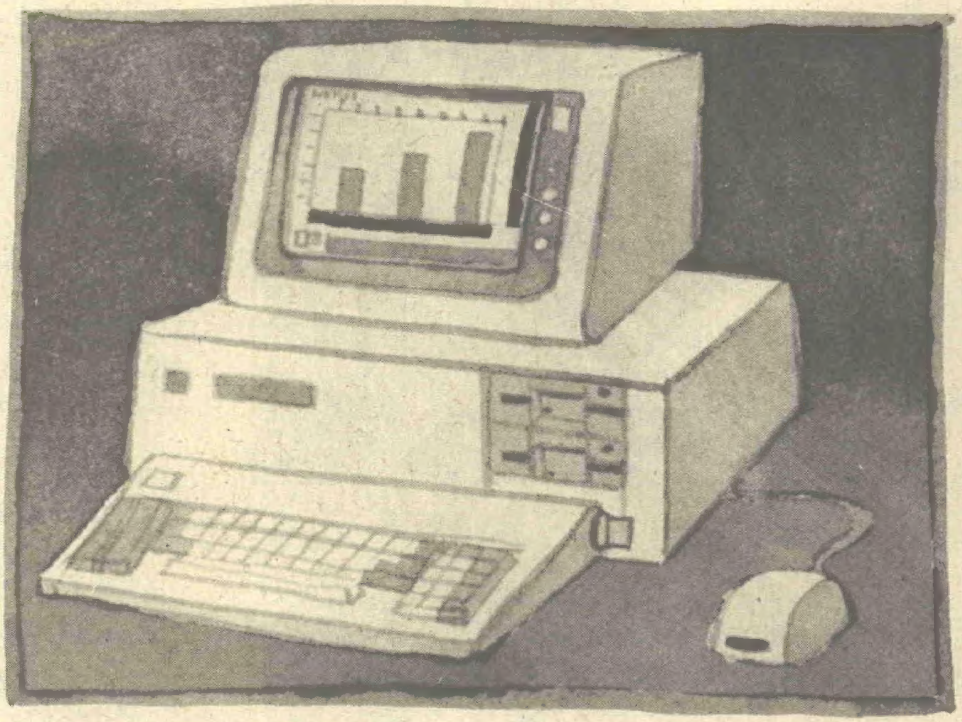
อาจหาญ สัตยารักษ์

SOFT MAIL

เมนู 3 มิติ กับเม้าส์

ให้แนวความคิดในเรื่องการเขียนกราฟิกใหม่ ๆ หลายประการเพื่อให้มีรูปแบบที่แปลกตายิ่งขึ้น



หลังจากที่ผู้เขียนได้เสนอบทความเรื่อง "สารบัญไฟล์แบบ 3 มิติ" ในฉบับที่ ๑๑ ไปแล้ว ปรากฏว่าผู้อ่านหลายท่านได้มีจดหมายไปถึงผู้เขียน เพื่อขอให้เขียนการเลือกเมนูแบบ 3 มิติบ้าง ในช่วงแรก ๆ ผู้เขียนก็ชี้แจงว่า ได้เขียนบทความประเภทนี้ไปบ้างแล้ว ขอให้ดัดแปลงโปรแกรมโดยยึดการแสดงแป้น 3 มิติจากบทความอีกที แต่เมื่อมีจดหมายพูดถึงเรื่องนี้หลายฉบับมากเข้า ผู้เขียนจึงตัดสินใจเขียนเรื่องเมนู 3 มิติ ดังที่ผู้อ่านจะได้ศึกษาต่อไป

เพื่อให้เกิดความแปลกใหม่สมกับการที่ วารสารไมโครคอมพิวเตอร์เป็นวารสารชั้นนำฉบับหนึ่งในเมืองไทย ผู้เขียนจึงนำเอาเม้าส์มาเกี่ยวข้องกับการเลือกรายการในเมนู 3 มิติด้วย

ซึ่งทำให้ผู้อ่านได้รับสิ่งที่ดีกว่าที่ขอร้องผู้เขียนไป รวมทั้งยังมีเทคนิคที่น่าสนใจอีกหลายอย่างด้วย

ไฟล์ในบทความทั้งหมดเหมาะกับการรันบนจอสีหรือจอวีจีเอโมโนโครมเท่านั้น การรันบนจอโมโนโครมธรรมดาจะไม่สามารถแสดงความเป็น 3 มิติให้ปรากฏได้

ไฟล์ที่ผู้อ่านจะได้รับจากบริการซอฟต์เมล์นั้นมี 4 ไฟล์คือ thai.tpu, head.pas, menu__ nol.pas, และ menu__ o2.pas

สำหรับบริการซอฟต์เมล์นี้ เป็นบริการที่ทางวารสารฯจัดทำขึ้นเพื่อผู้อ่านที่ไม่มีเวลาในการคีย์โปรแกรม หรือเพื่อให้มีความสมบูรณ์ในการศึกษาความรู้ตามบทความ โดยที่มิได้มุ่งหวังด้านกำไรแต่อย่างใด รวมทั้งบริการนี้ผู้เขียนบทความทุกท่านก็ไม่ได้รับค่าตอบแทนใด ๆ เช่นกัน จึงเป็นบริการเพื่อท่านผู้อ่านอย่างแท้จริง

แป้น 3 มิติ

เพื่อประโยชน์สูงสุดในการศึกษางานด้านกราฟิก ผู้เขียนจึงขอแจ้งให้ผู้อ่านทราบว่า การเขียนโปรแกรมในเรื่องแป้น 3 มิติและการใช้เม้าส์นี้ ได้เขียนในลักษณะยูนิตไฟล์โดยผู้เขียนจะใช้ชื่อว่า head.tpu

หลักการของการแสดงแป้น 3 มิตินั้นมีดังนี้

1. กำหนดพื้นส่วนกลางเป็นรูปสี่เหลี่ยม เพื่อแสดงเป็นแป้น

2. กำหนดพื้นส่วนขอบด้านซ้ายและด้านบนเป็นแบบเดียวกัน โดยส่วน

นี้จะแสดงด้วยสีหมายเลข 8 เพื่อให้เหมือนว่าเป็นเงาของแป้น

3. กำหนดพื้นส่วนขอบด้านขวาและด้านล่างเป็นแบบเดียวกัน โดยส่วนนี้แสดงด้วยสีหมายเลข 15 เพื่อให้ดูเหมือนว่าเป็นด้านที่ถูกแสงสว่างส่องกระทบ

ถ้าสลับสีของข้อ 2 และ 3 แป้นที่ปรากฏบนจอภาพจะดูเสมือนเป็นแป้นที่ยุบลงจากพื้นราบอื่น ๆ ซึ่งเมื่อเขียนโปรแกรมจะได้ดังรูปที่ 1 ซึ่งมีรายละเอียดดังนี้

1. บรรทัดที่ 1-3 กำหนดเป็นการเขียนยูนิตไฟล์ โดยเรียกใช้ยูนิต thai.tpu มาร่วมด้วย

2. 4-6 กำหนดตัวแปรใช้งาน โดย XMouse, YMouse ใช้บอกตำแหน่งของเม้าส์ ส่วนตัวแปร MaxColumn เป็นจำนวนคอลัมน์สูงสุดของรายการในเมนู ทั้งนี้เพราะว่าภาษาไทยนั้นมี 3 ระดับ จึงหาความกว้างของแป้นด้วยความยาวของข้อความไม่ได้ ซึ่งตัวแปรทั้งหมดสามารถใช้งานได้จากนอกยูนิตไฟล์

3. บรรทัดที่ 7-14 ชุดคำสั่งที่สามารถเรียกใช้จากภายนอกยูนิตไฟล์ได้

4. บรรทัดที่ 17-46 ชุดคำสั่งแสดงข้อความบนแป้น 3 มิติ ขนาดความกว้างของแป้นจะขึ้นกับตัวแปร MaxColumn ส่วนความสูงจะใช้ขนาด 28 จุดในแนวดิ่ง

5. บรรทัดที่ 25 วาดกรอบสีดำล้อมแป้นเพื่อให้แยกออกจากส่วนของพื้นอื่น

6. บรรทัดที่ 26 วาดพื้นของแป้นด้วยสีตามตัวแปร c1 ทำให้ได้รูปสี่เหลี่ยมทึบ

7. บรรทัดที่ 21-29 วาดเงาของแป้นด้านบนและด้านซ้าย

8. บรรทัดที่ 30-38 วาดเงาของแป้นด้านล่างและด้านขวา

9. บรรทัดที่ 39 แสดงข้อความให้เป็นสีตามตัวแปร c2

10. บรรทัดที่ 41-63 การวาดกระดาน 3 มิติด้วยพื้นสีตามตัวแปร CPatt โดยมีเงาเป็นสีตามแนวคิดที่ได้กล่าวมาแล้ว

รูปที่ 1 ยูนิต head.tpu แสดงข้อความบนแป้น 3 มิติและกระดานแบบ 3 มิติ

```
 1: UNIT Head;
 2: interface
 3: uses crt,dos,graph,thai;
 4: Var Mouse:boolean;    XMouse,YMouse,MaxColumn:integer;
 5: PROCEDURE Block3D(x1,y1:integer; Key:string; C1,C2,Mode:byte);
 6: PROCEDURE Board3D(x1,y1,x2,y2,Cpatt,Mode:word);
 7: PROCEDURE MouseOn;
 8: PROCEDURE MouseOff;
 9: PROCEDURE SetPosition(x,y:word);
10: PROCEDURE WindowMouse(XLeft,Yup,Xright,Ydown:word);
11: PROCEDURE FindXY(Var x,y:integer);
12: PROCEDURE ReadMouse;
13: implementation

14: var Box3D:array[1..6] of PointType;
15: PROCEDURE Block3D(x1,y1:integer; Key:string; C1,C2,Mode:byte);
16: var  x2,y2:word;
17: begin
18:   x2:=x1+MaxColumn*9; y2:=y1+27;
19:   setcolor(0);        rectangle(x1-1,y1-1,x2+1,y2);
20:   SetFillStyle(1,C1); Bar(x1+2,y1+2,x2-2,y2-2);
21:   Box3D[1].x:=x1;     Box3D[1].y:=y1;
22:   Box3D[2].x:=x2;     Box3D[2].y:=y1;
23:   Box3D[3].x:=x2-2;   Box3D[3].y:=y1+2;
24:   Box3D[4].x:=x1+2;   Box3D[4].y:=y1+2;
25:   Box3D[5].x:=x1+2;   Box3D[5].y:=y2-2;
26:   Box3D[6].x:=x1;     Box3D[6].y:=y2;
27:   if Mode=1 then SetFillStyle(1,8)
28:             else  SetFillStyle(1,15);
29:   FillPoly(6,Box3D);
30:   Box3D[1].x:=x1;     Box3D[1].y:=y2;
31:   Box3D[2].x:=x1+2;   Box3D[2].y:=y2-2;
32:   Box3D[3].x:=x2-2;   Box3D[3].y:=y2-2;
33:   Box3D[4].x:=x2-2;   Box3D[4].y:=y1+2;
34:   Box3D[5].x:=x2;     Box3D[5].y:=y1;
35:   Box3D[6].x:=x2;     Box3D[6].y:=y2;
36:   if Mode=1 then  SetFillStyle(1,15)
37:             else  SetFillStyle(1,8);
38:   FillPoly(6,Box3D);
39:   setcolor(C2);  OutThaiXY(x1+10,y1+2,#17+Key);
40: end;

41: PROCEDURE Board3D(x1,y1,x2,y2,Cpatt,Mode:word);
42: begin
43:   setcolor(0); rectangle(x1-1,y1-1,x2+1,y2);
44:   SetFillStyle(1,CPatt);  Bar(x1,y1,x2,y2);
45:   Box3D[1].x:=x1;     Box3D[1].y:=y1;
46:   Box3D[2].x:=x2;     Box3D[2].y:=y1;
```

รูปที่ 1 (ต่อ)

```
47:   Box3D[3].x:=x2-2;   Box3D[3].y:=y1+2;
48:   Box3D[4].x:=x1+2;   Box3D[4].y:=y1+2;
49:   Box3D[5].x:=x1+2;   Box3D[5].y:=y2-2;
50:   Box3D[6].x:=x1;     Box3D[6].y:=y2;
51:   if Mode=1 then SetFillStyle(1,8)
52:             else  SetFillStyle(1,15);
53:   FillPoly(6,Box3D);
54:   Box3D[1].x:=x1;     Box3D[1].y:=y2;
55:   Box3D[2].x:=x1+2;   Box3D[2].y:=y2-2;
56:   Box3D[3].x:=x2-2;   Box3D[3].y:=y2-2;
57:   Box3D[4].x:=x2-2;   Box3D[4].y:=y1+2;
58:   Box3D[5].x:=x2;     Box3D[5].y:=y1;
59:   Box3D[6].x:=x2;     Box3D[6].y:=y2;
60:   if Mode=1 then SetFillStyle(1,15)
61:             else  SetFillStyle(1,8);
62:   FillPoly(6,Box3D);
63: end;
```

เม้าส์กับการทำงาน

เพื่อให้การเลือกรายการในเมนูทันสมัยมากขึ้น จึงจำเป็นต้องอาศัยความรู้เกี่ยวกับเม้าส์มาทำงานด้วย โดยจะอาศัยอินเตอร์รัพต์หมายเลข 33H มาช่วยงาน การทำงานเกี่ยวกับเม้าส์นั้นผู้อ่านต้องทำดังนี้

1. ต้องตรวจว่าได้เสียบสายของเม้าส์เข้าที่ด้านหลังเครื่องคอมพิวเตอร์ในช่องที่เขียนว่า COM1 โดยปกติเราคงเสียบคาทิ้งไว้แล้ว

2. รันไฟล์ไดรเวอร์ของเม้าส์ ซึ่งเราจะได้มาพร้อมกับการซื้อเม้าส์มาใช้ ในกรณีที่ผู้อ่านเสียบเม้าส์ที่ช่อง COM2 จะต้องระบุในการรันไฟล์ดังกล่าวนี้ด้วย โดยอ่านจากคู่มือของเม้าส์ที่เราซื้อมา

3. รันไฟล์ที่ทำงานร่วมกับเม้าส์ได้ ซึ่งจะทำให้เราได้ใช้งานเม้าส์ตามที่โปรแกรมกำหนดไว้

ส่วนการเขียนโปรแกรมจริง ๆ นั้น เราจะต้องให้โปรแกรมทำงานได้ทั้งการใช้แป้นและการใช้เม้าส์ด้วย การทำงานเกี่ยวกับเม้าส์นั้นเราจะต้องทราบฟังก์ชันต่าง ๆ เพื่อที่จะสามารถกำหนดให้เม้าส์ปรากฏหรือไม่ก็ได้ รวมทั้งสามารถตรวจหาตำแหน่งของเม้าส์ การกดปุ่มของเม้าส์ ฯลฯ

การทำงานของเม้าส์ไดรเวอร์นั้นจะทำการจำภาพหน้าจอและคืนภาพหน้าจอตลอดเวลาที่เราเคลื่อนที่เม้าส์ไปยังส่วนต่าง ๆ ของจอภาพ ดังนั้นคำสั่งยกเลิกการแสดงเคอร์เซอร์เม้าส์ก็คือการคืนภาพหน้าจอด้วย

ซึ่งจะเขียนโปรแกรมตรวจสอบและใช้งานได้ดังรูปที่ 2 มีรายละเอียดคือ

1. บรรทัดที่ 64-71 ตรวจว่าได้รันไฟล์ไดรเวอร์ของเม้าส์แล้วหรือยัง โดยกำหนดฟังก์ชัน 0 และใช้อินเตอร์รัพต์หมายเลข 33H ซึ่งทำให้ได้ค่าตัวแปร regs คืนกลับมา โดยจะได้ค่า regs.AX ไม่เท่ากับ 0 ซึ่งหมายถึงได้ติดตั้งเม้าส์แล้ว

2. บรรทัดที่ 72-73 กำหนดฟังก์ชัน 1 (regs.AX=1) เพื่อแสดงเคอร์เซอร์ของเม้าส์

3. บรรทัดที่ 74-75 กำหนดฟังก์ชัน 2 เพื่อหยุดแสดงเคอร์เซอร์ของเม้าส์

4. บรรทัดที่ 76-80 ให้ค่าแก่

รูปที่ 2 ชุดคำสั่งเกี่ยวกับการใช้เม้าส์และการรอกดปุ่มของเม้าส์

```
64: PROCEDURE CheckMouse;
65: begin
66:  Mouse:=false; regs.AX:=0; Intr($33,regs);
67:  if regs.AX<>0 then
68:   begin
69:    Mouse:=True; WindowMouse(0,0,getmaxx-8,getmaxy-8);
70:   end;
71: end;

72: PROCEDURE MouseOn;
73: begin regs.AX:=1; Intr($33,regs); end;

74: PROCEDURE MouseOff;
75: begin regs.AX:=2; Intr($33,regs); end;

76: PROCEDURE SetPosition(x,y:word);
77: begin
78:   regs.CX:=x; regs.DX:=y;
79:   regs.AX:=4; Intr($33,regs);
80: end;
```

รูปที่ 2 (ต่อ)

```
81: PROCEDURE WindowMouse(XLeft,Yup,Xright,Ydown:word);
82: begin
83:   regs.CX:=XLeft; regs.DX:=Xright;
84:   regs.AX:=7;     Intr($33,regs);
85:   regs.CX:=Yup;   regs.DX:=YDown;
86:   regs.AX:=8;     Intr($33,regs);
87: end;

88: PROCEDURE FindXY(Var x,y:integer);
89: begin
90:  regs.AX:=4;  Intr($33,regs);
91:  x:=regs.CX;  y:=regs.DX;
92: end;

93: PROCEDURE ReadMouse;
94: begin
95:  if Mouse then
96:   begin
97:    WindowMouse(0,0,getmaxx-8,getmaxy-8);
98:    delay(300);  Regs.BX:=0;
99:   end;
100:  repeat
101:   if Mouse then
102:    begin
103:      MouseOn; regs.AX:=3; Intr($33,regs);
104:      if Regs.BX in [1..2] then
105:        begin  Mouseoff;  Exit;   end;
106:    end;
107:  until keypressed;
108:   readchar;
109: end;

110: begin
111:   CheckMouse;
112: end.
```

ตัวแปร regs และเรียกใช้ฟังก์ชัน 4 เมื่อกำหนดตำแหน่งของเคอร์เซอร์เม้าส์

5. บรรทัดที่ 81-87 กำหนดฟังก์ชัน 7 เพื่อกำหนดขอบเขตของเม้าส์ในแนวนอน (แกน x) และกำหนดฟังก์ชัน 8 (regs.AX=8) เพื่อกำหนดขอบเขตเม้าส์ในแนวตั้ง (แกน y)

6. บรรทัดที่ 88-92 กำหนดฟังก์ชัน 4 เพื่อหาตำแหน่งของเม้าส์ในแกน x และแกน y โดยเป็นการกำหนดให้กับตัวแปร x และ y ซึ่งค่านี้จะคืนไปยังค่าที่เรียกเข้ามา

7. บรรทัดที่ 93-109 ส่วนการเรียกใช้ฟังก์ชัน 3 (regs.AX=3) เพื่อตรวจการกดปุ่มของเม้าส์ ซึ่งจะได้ค่า Rags.BX เป็น 1 ถ้ากดปุ่มซ้ายของเม้าส์ และมีค่าเป็น 2 ถ้ากดปุ่มขวาของเม้าส์

8. บรรทัดที่ 95-99 กำหนดขอบเขตพื้นที่ที่เม้าส์สามารถเคลื่อนย้ายไปมาได้ การหน่วงเวลาจะช่วยป้องกันการกดปุ่มแช่ไว้นานเกินไป

9. บรรทัดที่ 100-107 โปรแกรมจะวนรอบรอการกดปุ่มของเม้าส์หรือกดแป้นพิมพ์ ซึ่งจะออกจากชุดคำสั่งนี้ได้ทั้งสองกรณี

โปรแกรมทั้งรูปที่ 1 และ 2 นี้จะอยู่ในไฟล์ชื่อ head.pas ตามชื่อยูนิตเท่านั้น

ผู้เขียนขอให้ผู้อ่านลองย้อนกลับไปดูโปรแกรมบรรทัดที่ 69 และบรรทัดที่ 81-87 อีกครั้ง และให้ลองคิดดูว่าการที่ผู้เขียนกำหนดอย่างนี้ ทำให้เกิดความคิดอะไรบ้าง

จุดประสงค์ที่ผู้เขียนต้องการให้ผู้อ่านมองเห็นในจุด 2 จุดที่แจ้งให้ดูข้างต้นจะเป็นดังนี้

1. บรรทัดที่ 69 เรียกใช้ชุดคำสั่งของบรรทัดที่ 81-87 อันเป็นชุดคำสั่งที่อยู่ตามหลังการเรียกใช้ แต่ปรากฏว่าไฟล์นี้กลับสามารถผ่านการคอมพายล์ได้

2. คำตอบที่ผู้อ่านคงรู้แล้วก็คือชุดคำสั่งและฟังก์ชันที่เขียนในยูนิตนี้ไม่จำเป็นต้องเรียงลำดับกันก็ได้ ซึ่งจะต่างกับการเขียนในไฟล์ปกติที่จะต้องเรียกชุดคำสั่งหรือฟังก์ชันที่ตามหลังไม่ได้

3. ในยูนิตไฟล์เมื่อถูกเรียกใช้การทำงานจะเป็นการตรวจสอบทั้งชุดของยูนิต จึงทำให้สามารถเรียกใช้ชุดคำสั่งหรือฟังก์ชันที่อยู่ในส่วนใดก็ได้

เมนูแบบที่หนึ่ง

ตอนนี้เราก็มียูนิตใช้งานที่พร้อมจะเขียนเมนู 3 มิติได้แล้ว โดยมี thai.tpu และ head.tpu ดังนั้นมาเริ่มศึกษาการเขียนเมนูแบบ 3 มิติกันต่อไป เพื่อให้ผู้อ่านได้รับประโยชน์สูงสุดในการนำไปใช้งาน ผู้เขียนจึงกำหนดแนวคิดไว้ดังนี้

1. จะเขียนโปรแกรมตัวอย่าง 2 โปรแกรม โดยแบบแรกเป็นแบบมีรายการเลือกเพียงชุดเดียว และแบบที่สองจะมีรายการเลือกมากกว่า 1 ชุด (ผู้เขียน

เลือกตัวอย่างแบบมีรายการเลือก 3 ชุด)

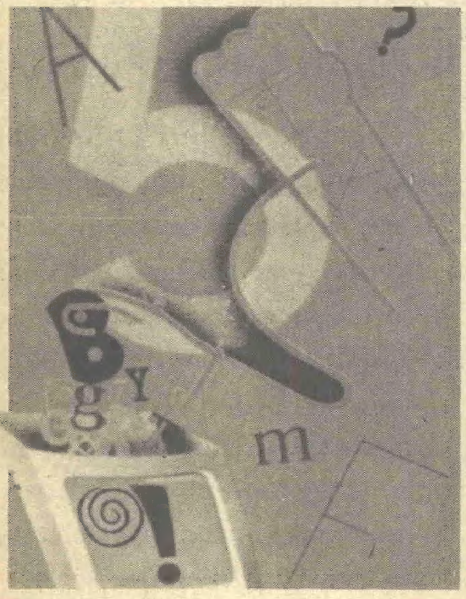
2. ทุกโปรแกรมจะต้องสามารถเปลี่ยนตำแหน่งของเมนูได้โดยง่าย คือใช้เพียงการกำหนดค่าตัวแปรเพียงไม่กี่ค่าเท่านั้น โดยไม่ต้องแก้ไขส่วนอื่น ๆ ของโปรแกรมเลย

3. ทั้งสองโปรแกรมต้องมีความแตกต่างกันในการเขียนโปรแกรม

4. ผู้อ่านสามารถนำไปใช้งานได้โดยง่าย เพียงกำหนดข้อความในรายการใหม่และกำหนดค่าตัวแปรที่เกี่ยวข้องเพื่อกำหนดตำแหน่งของเมนูเท่านั้น ก็จะได้รูปแบบเมนู 3 มิติที่แตกต่างกันออกไป

5. การเลือกรายการในเมนูจะทำได้ 2 แบบคือ การใช้แป้นพิมพ์และการใช้เม้าส์

สำหรับในตอนแรกนี้ขอเสนอเมนู 3 มิติแบบที่มีกลุ่มรายการเลือกเพียงกลุ่มเดียวก่อน ซึ่งโดยลักษณะนี้จะเป็นการแสดงรายการเรียงจากบนลงล่างให้แยกออกจากกัน การทำงานในกรณีใช้แป้นพิมพ์จะใช้แป้นลูกศรขึ้นและลูกศรลงมาช่วยเลื่อนแถบสว่าง ส่วนการใช้เม้าส์นั้นเราใช้วิธีตรวจสอบขอบเขตของ



ตำแหน่งเม้าส์ว่าเป็นตำแหน่งที่รายการในเมนูหรือไม่

ตามแนวคิดที่ว่าโปรแกรมต้องสามารถย้ายตำแหน่งของเมนูให้ไปแสดงบนจอภาพ ณ ตำแหน่งใดก็ได้นั้น เราจำเป็นต้องเขียนโปรแกรมให้มีลักษณะดังนี้

1. ให้แสดงรายการเมนู 3 มิติทั้งหมดที่บริเวณมุมบนซ้ายของจอภาพ คือให้มุมบนซ้ายของเมนูอยู่ที่จุด 0,0 (กราฟิก)

2. เขียนโปรแกรมทดสอบการทำงานต่าง ๆ เช่น การใช้แป้นพิมพ์เลือกรายการให้ทำงานได้ถูกต้องที่มุมบนซ้ายนี้ด้วย

3. เมื่อต้องการย้ายรายการเมนูและการทำงานในการเลือกเมนูไปที่ตำแหน่งอื่น เราก็เพียงกำหนดขอบเขตของจอภาพด้วยคำสั่ง setviewport ใหม่เท่านั้น เมนูและการทำงานทั้งหมดจะสามารถย้ายไปยังที่ใหม่ได้โดยไม่ต้องแก้ไขโปรแกรมตามข้อ 1 และ 2 แต่อย่างใด

4. ปัญหาที่เกิดกับการเขียนโปรแกรมแบบใช้เม้าส์คือ คำสั่ง setviewport ไม่มีผลต่อเม้าส์ คือ ค่าตำแหน่งของเม้าส์ที่หาได้จาก findxy ยังคงให้ค่าจริงของจอภาพ ดังนั้นเราจึงต้องตรวจตำแหน่งของเม้าส์ให้ตรงกับตำแหน่งของรายการในเมนูที่ย้ายที่ไปด้วย ซึ่งผู้เขียนใช้แนวคิดดังนี้

4.1 กำหนดตัวแปร XMenu และ YMenu เพื่อใช้กำหนดตำแหน่งใหม่ของเมนู

4.2 หาตำแหน่งของเม้าส์ให้กับตัวแปร XMouse และ YMouse

4.3 ตำแหน่งของเม้าส์เทียบกับเมนูใหม่จึงหาได้จากคำสั่ง Dec (XMouse, XMenu) สำหรับค่าในแกน x และ Dec (YMouse,YMenu) สำหรับค่าในแกน y ซึ่งเราก็จะสามารถนำค่า XMouse และ YMouse ไปตรวจสอบได้ว่า การกดปุ่มของเม้าส์นั้นอยู่ในขอบเขตของรายการในเมนูหรือไม่ (คือถ้ากดปุ่มของเม้าส์ที่ส่วนของจอภาพที่ไม่ใช่รายการในเมนูก็ไม่มีการเลือกรายการ)

นำแนวคิดนี้มาเขียนโปรแกรมเพื่อแสดงรายการในเมนูเลือกและส่วนการเลือกรายการในเมนูได้ตามรูปที่ 3 โดยผู้เขียนให้แนวคิดสำหรับภาษาไทยรหัสเกษตรที่ป็อนได้จากราชวิถีเวิร์ดพีซี

รายละเอียดของโปรแกรมในรูปที่ 3 มีดังนี้

1. บรรทัดที่ 1 กำหนดยูนิตที่ใช้งานในโปรแกรมรวม 5 ยูนิต

2. บรรทัดที่ 2-4 กำหนดตัวแปรใช้งาน คือ

- MaxMenu จำนวนสูงสุดของรายการในเมนู (1-10)
- Choice รองรับลำดับรายการเมนูที่เลือก
- MenuST ข้อความรองรับรายการเมนู
- XMenu, YMenu ตำแหน่งมุมบนซ้ายของเมนูที่เราต้องการให้แสดงรายการเมนู 3 มิติ

3. บรรทัดที่ 5-25 ชุดคำสั่ง showmenu คือส่วนการแสดงเมนูและรายการในเมนู

4. บรรทัดที่ 8-16 คำนวณหาค่า MaxColumn เพื่อที่ให้แป้น 3 มิติมีความกว้างมากพอที่จะแสดงรายการในเมนูทั้งหมดได้อย่างสวยงาม (คือจะได้ความกว้างเท่ากับข้อความที่ยาวที่สุดทุกแป้น)

5. บรรทัดที่ 17 แสดงกระดาน 3 มิติในขนาดครอบคลุมรายการเมนูทั้งหมดได้ ซึ่งตามคำสั่งจะเป็นการแสดง

รูปที่ 3
การแสดงข้อความของรายการในเมนูและการเลือกรายการในเมนู

```
 1: uses crt,dos,graph,thai,Head;
 2: var MaxMenu,Choice:byte;
 3:     MenuST:array[1..10] of string;
 4:     XMenu,Ymenu:word;

 5: PROCEDURE ShowMenu;
 6: var Column,k,p:byte;
 7: begin
 8:  MaxColumn:=0;
 9:  for k:=1 to MaxMenu do
10:   begin
11:    Column:=0;
12:    for p:=1 to length(MenuST[k]) do
13:        if MenuST[k][p] in [#32..#214] then Inc(Column);
14:    if MaxColumn<Column then MaxColumn:=Column;
15:   end;
16:   inc(MaxColumn,2);
17:   Board3D(0,0,MaxColumn*9+20,MaxMenu*30+13,7,1);
18:  for k:=1 to MaxMenu do
19:    Block3D(10,k*30-22,MenuST[k],12,14,0);
20:  if Mouse then
21:   begin
22:     WindowMouse(10+Xmenu,8+YMenu,MaxColumn*9+8+Xmenu,MaxMenu*30+2+Ymenu);
23:     SetPosition((20+MaxColumn*9) div 2+Xmenu,12+Ymenu); MouseOn;
24:   end;
25: end;

26: PROCEDURE SlideMenu;
27: const Ypix:array[0..10] of word=
28:         (8,38,68,98,128,158,188,218,248,278,308);

29:       Procedure Checkchoice;
30:       begin
31:         choice:=0; Dec(YMouse,YMenu);
32:         if Ymouse>Ypix[0] then
33:          begin
34:            Repeat
35:              inc(choice);
36:            until Ypix[choice]>Ymouse;
37:          end;
38:          MouseOff;
39:          Block3D(10,choice*30-22,MenuST[Choice],4,14,1);
40:          MouseOn;
41:       end;

42: begin
43:   ShowMenu;     choice:=1;
44:   if Mouse then MouseOn;
45:  Repeat
46:    repeat
47:      if Mouse then
48:        begin
49:          regs,AX:=3; Intr($33,regs);
50:          if Regs.BX in [1..2] then
51:            begin
52:              Findxy(XMouse,YMouse);
53:             if (YMouse-YMenu-8) mod 30 in [2..24] then
54:              begin
55:                Block3D(10,choice*30-22,MenuST[Choice],12,14,0);
```

รูปที่ 3 (ต่อ)

```
56:                Checkchoice;
57:                if choice in [1..MaxMenu] then
58:                 begin
59:                   MouseOff; Exit;
60:                 end;
61:               end;
62:             end;
63:         end;
64:  until keypressed;
65:   readchar;
66:   if Mouse then MouseOff;
67:   if ch<>#13 then
68:          Block3D(10,choice*30-22,MenuST[Choice],12,14,0);
69:   case funckey of
70:    #72:begin
71:          if choice>1 then Dec(Choice) Else Choice:=MaxMenu;
72:          Block3D(10,choice*30-22,MenuST[Choice],4,14,1);
73:        end;
74:    #80:begin
75:          if choice<MaxMenu then Inc(Choice) Else Choice:=1;
76:          Block3D(10,choice*30-22,MenuST[Choice],4,14,1);
77:        end;
78:   end;
79:   if Mouse then
80:    begin
81:     SetPosition((20+MaxColumn*9) div 2+Xmenu,12+Ymenu+(choice-1)*30);
82:     MouseOn;
83:    end;
84: until ch=#13;
85:  if Mouse then MouseOff;
86: end;
```

ที่มุมบนซ้ายของจอภาพ

6. บรรทัดที่ 18-19 แสดงข้อความในรายการของเมนูบนแป้น 3 มิติ ด้วยแป้นสีแดงใส (สีหมายเลข 12) ข้อความสีเหลือง (สีหมายเลข 14) โดยเป็นแป้นนูน (ตัวแปรเป็น 0) ตามคำสั่งก็ยังเป็นการแสดงที่มุมบนซ้ายเช่นกัน ตำแหน่งบนสุดของแป้นแรกจะอยู่ที่ 1×30-22=8 และแต่ละแป้นจะมีตำแหน่งบนสุดของแป้นห่างกัน 30 จุด ทำให้ตำแหน่งล่างสุดของแป้นแรกอยู่ที่จุด 8+30=38 ซึ่งเราจะนำไปใช้ในการตรวจสอบว่าผู้ใช้กดปุ่มของเม้าส์เมื่ออยู่ที่รายการใด

7. บรรทัดที่ 20-24 กำหนดการทำงานเมื่อพบว่าได้ติดตั้งเม้าส์ไดรเวอร์ไว้ โดยเป็นการกำหนดขอบเขตที่จะให้เม้าส์สามารถเคลื่อนที่ได้ ซึ่งตามคำสั่งในบรรทัดที่ 22 นั้นผู้เขียนได้กำหนดให้เม้าส์เคลื่อนที่ได้เฉพาะในกรอบเมนูเท่านั้น และกำหนดตำแหน่งแรกให้เม้าส์แสดง ณ แป้นแรกในรายการเมนู (ขอให้สังเกตการนำตัวแปร XMenu, YMenu มาใช้ในการกำหนดค่าทั้งสองคำสั่งด้วย)

ประโยชน์ที่ได้จากการกำหนดขอบเขตของเม้าส์ไว้ในรายการเมนูแบบนี้ จะช่วยให้เราไม่ต้องตรวจสอบว่าเมื่อกดปุ่มของเม้าส์นั้นค่าในแกน x อยู่ในขอบเขตเมนูหรือไม่ ทั้งนี้เพราะมีรายการเมนูเพียงชุดเดียว

8. บรรทัดที่ 26-86 ชุดคำสั่ง slidemenu เพื่อทำการเลือกรายการในเมนู

9. บรรทัดที่ 27 กำหนดค่าคงที่เพื่อใช้ตรวจสอบว่าผู้ใช้กดปุ่มเม้าส์ที่รายการใด ซึ่งจะเป็นด้านล่างสุดของแป้นแต่ละแป้น

10. บรรทัดที่ 29-41 ส่วนการตรวจว่ากดปุ่มของเม้าส์ที่รายการใด

11. บรรทัดที่ 31 กำหนดรายการเริ่มต้น และทำให้ค่าของ YMouse เป็นค่าจริงเมื่ออยู่มุมบนซ้าย

12. บรรทัดที่ 32-37 ตรวจสอบค่า YMouse เทียบกับ Ypix[choice] ซึ่งทำให้เราทราบได้ว่าอยู่ในรายการใดของเมนู

13. บรรทัดที่ 38-40 แสดงรายการเมนูที่ตรวจได้ใหม่ในลักษณะแป้น

เว้าลง

ขอให้สังเกตด้วยว่าทุก ๆ ครั้งที่เราจะเขียนภาพหรือข้อความลงบนจอภาพใหม่ ต้องใช้คำสั่ง mouseoff ก่อน เพราะคำสั่งนี้เม้าส์ไดรเวอร์จะคืนภาพหน้าจอให้กับจอภาพ เมื่อเขียนภาพหรือข้อความเสร็จจึงกำหนดให้ mouseon ใหม่ ซึ่งเม้าส์ไดรเวอร์จะเก็บภาพหน้าจอไว้อีกครั้ง

14. บรรทัดที่ 43-44 แสดงรายการในเมนู และกำหนดให้แสดงเคอร์เซอร์เม้าส์

15. บรรทัดที่ 46-64 ส่วนการรอกดแป้นหรือกดปุ่มของเม้าส์

16. บรรทัดที่ 47-63 ส่วนการทำงานด้วยเม้าส์

17. บรรทัดที่ 49 แสดงเม้าส์ตามการเลื่อนเม้าส์ของผู้ใช้

18. บรรทัดที่ 52 เมื่อกดปุ่มซ้ายหรือปุ่มขวาของเม้าส์ จะหาตำแหน่งของเม้าส์

19. บรรทัดที่ 53 ตรวจว่าบริเวณที่จะมีผลในการกดปุ่มของเม้าส์อยู่ที่ขอบบนของแป้นถึงขอบล่างของแป้นเท่านั้น (กดที่บริเวณเงาบน บริเวณเงาล่าง หรือบริเวณอื่น ๆ จะไม่มีผล)

20. บรรทัดที่ 55 แสดงแป้น ณ ตำแหน่งเดิมให้เป็นแบบแป้นนูน (กำหนดไว้เผื่อการกดแป้นลูกศรเลื่อนรายการในเมนู)

21. บรรทัดที่ 56-60 ตรวจสอบว่าเป็นการเลือกรายการใด ซึ่งถ้าอยู่ในรายการที่กำหนดจะออกจากชุดคำสั่งนี้ไป

22. บรรทัดที่ 65-85 ส่วนการทำงานเมื่อกดแป้นลูกศรขึ้นหรือลง ซึ่งคงทำความเข้าใจได้ด้วยตนเอง

ชุดคำสั่งทั้งหมดเมื่อรันบนจอภาพจะสามารถเปลี่ยนตำแหน่งรายการเมนูได้ โดยชุดคำสั่งในรูปที่ 3 นี้จะทำงานได้โดยไม่ต้องแก้ไขโปรแกรมแม้แต่บรรทัดเดียว

ส่วนการแสดงข้อความในเมนูตลอดจนถึงการกำหนดตำแหน่งของเมนูจะเป็นดังโปรแกรมในรูปที่ 4 ซึ่งรายละเอียดของโปรแกรมมีดังนี้

1. บรรทัดที่ 90 ก่อนที่วาดภาพหรือแสดงข้อความใด ๆ ต้องกำหนด

```
 87: PROCEDURE MainMenu;
 88: begin
 89:  Repeat
 90:   if Mouse then Mouseoff;
 91:   Board3D(0,0,639,348,3,0);
 92:   Board3D(30,10,609,60,9,1);
 93:   OutThaixy(165,13,#17#3'6'#1'เมนู 3 มิติเลือกรายการด้วยเมาส์หรือแป้นพิมพ์(1)');
 94:   OutThaixy(180,35,#17'เขียนโปรแกรมด้วย นายอาจหาญ สัตยารักษ์');
 95:   MenuST[1]:='สร้างไฟล์เก็บข้อมูลนักเรียน';
 96:   MenuST[2]:='ป้อนรายชื่อนักเรียน';
 97:   MenuST[3]:='แก้ไขรายชื่อนักเรียน';
 98:   MenuST[4]:='เพิ่มรายชื่อนักเรียน';
 99:   MenuST[5]:='พิมพ์รายชื่อนักเรียน';
100:   MenuST[6]:=' =====เลิกงาน======';
101:   MaxMenu:=6;    XMenu:=200; YMenu:=100;
102:   setviewport(Xmenu,YMenu,getmaxx,getmaxy,clipon);
103:   SlideMenu;
104:   setviewport(0,0,getmaxx,getmaxy,clipon);
105:   block3D(210,300,MenuST[choice],9,15,0);
106:   Readmouse;
107:   case choice of
108:       1..10: {procedure for run}
109:   end;
110:  until choice=MaxMenu;
111: end;

112: {================== MENU_NO1.PAS ====================}
113: begin
114:   Mainmenu;
115:   closegraph;
116: end.
```

รูปที่ 4
ส่วนของโปรแกรมหลักของเมนูแบบรายการเดียว

ให้หยุดการแสดงเคอร์เซอร์เม้าส์ก่อนด้วยทุกครั้ง

2. บรรทัดที่ 91-94 แสดงกระดาน 3 มิติแบบนูนเต็มจอภาพ และแบบเว้าในขอบเขตที่แสดงข้อความในบรรทัดที่ 94-94 ได้พอดี

3. บรรทัดที่ 95-100 กำหนดข้อความที่จะเป็นรายการในเมนู

4. บรรทัดที่ 101 กำหนดจำนวนรายการในเมนูและตำแหน่งมุมบนซ้ายของเมนู (XMenu, YMenu) ซึ่งจะทำให้เมนูแสดงบนจอภาพได้ตามความต้องการ

5. บรรทัดที่ 102 กำหนดขอบเขตของจอภาพให้มีมุมบนซ้ายอยู่ที่ XMenu, YMenu ตามที่ได้กำหนด

6. บรรทัดที่ 103 เรียกใช้ชุดคำสั่ง slidemenu ซึ่งจะทำงานโดยยึดมุมบนซ้ายใหม่เป็นหลักในการตรวจสอบ ทำให้มีผลการทำงานถูกต้องเช่นเดิม

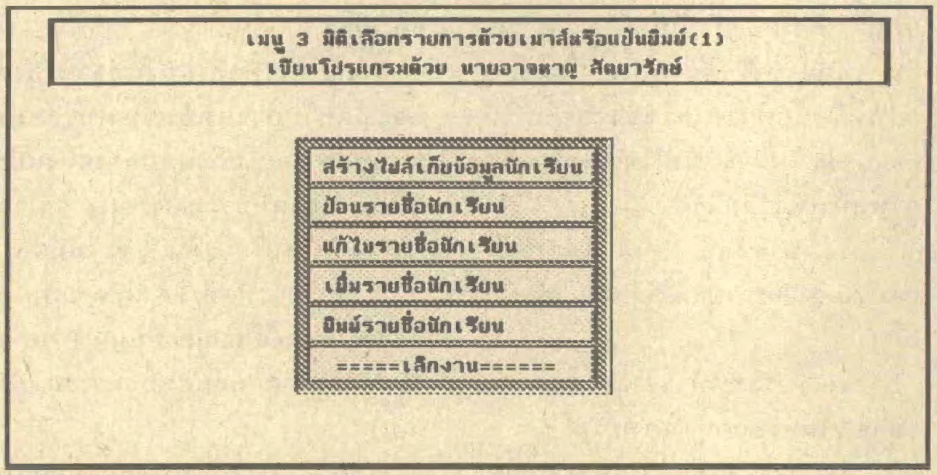
7. บรรทัดที่ 104-106 คำสั่งทั้งสามบรรทัดนี้ผู้เขียนต้องการเพียงให้เห็นว่าการเลือกรายการในเมนูนั้นเป็นไปอย่างถูกต้องหรือไม่เท่านั้น เมื่อใช้งานจริงก็ตัดออกไปได้

8. บรรทัดที่ 107-109 ส่วนที่ผู้พัฒนาโปรแกรมต้องเพิ่มเติมเองโดยแยกแต่ละกรณีออกจากกัน ในที่นี้ต้องการให้เห็นเป็นตัวอย่างเท่านั้น

9. บรรทัดที่ 110 การทำงานจะวนรอบเลือกเมนูอีก จนกว่าจะเลือกรายการสุดท้ายในเมนู

โปรแกรมทั้งหมดในรูปที่ 3 และ 4 เป็นโปรแกรมเดียวกัน โดยในซอฟต์เมล์คือไฟล์ menu__no1.pas เมื่อรันจะปรากฏผลบนจอภาพดังรูปที่ 5

เมื่อรันโปรแกรมดูผลบนจอภาพและทดสอบการทำงานที่การใช้แป้นลูกศรและการใช้เม้าส์แล้ว ขอให้ผู้อ่านลองเปลี่ยนค่า XMenu, YMenu ในบรรทัดที่ 101 แล้วรันดูผลใหม่ ก็จะเห็นว่าการย้ายส่วนของเมนูนี้เป็นเรื่องที่ทำได้ง่าย ๆ



รูปที่ 5 ภาพหน้าจอจากการรันไฟล์ menu__no1.pas

เมนูแบบที่สอง

ในบางครั้งเราจำเป็นต้องมีรายการในเมนูหลายชุด ซึ่งจะแตกต่างจากไฟล์ menu__no1.pas ที่มีอยู่เพียงชุดเดียวเท่านั้น ผู้เขียนจึงขอเสนอไฟล์ menu__no2.pas เป็นตัวอย่างของการมีรายการเมนูหลาย ๆ ชุด โดยตัวอย่างที่เขียนนี้จะมีรายการอยู่เพียง 3 ชุด และแต่ละชุดมีรายการเลือกไม่เท่ากัน

โดยหลักการเดิม ผู้เขียนขอเสนอโปรแกรมที่สามารถกำหนดตำแหน่งของเมนูทั้งสามให้อยู่ตรงส่วนใดของจอภาพก็ได้ โดยใช้วิธีกำหนดค่าให้กับตัวแปรเท่านั้น ซึ่งผู้เขียนหวังว่าจะช่วยปฏิวัติการแสดงเมนูแบบเดิมให้มาใช้แบบนี้แทน การทำงานของไฟล์ต่อไปนี้จะมีแนวคิดที่ต่างจากแบบแรกอยู่เรื่องหนึ่งคือ ระหว่างการเลือกรายการในเมนูนั้นจะใช้ขอบเขตของจอภาพเต็มจอตามปกติ ซึ่งแบบเดิมจะเป็นการกำหนดขอบเขตโดยยึดมุมบนซ้ายของเมนูเป็นหลัก

ซึ่งจะเขียนโปรแกรมเบื้องต้นได้ดังรูปที่ 6 มีรายละเอียดดังนี้

1. บรรทัดที่ 2 กำหนดให้รายการเมนูมี 3 ชุด

2. บรรทัดที่ 3-7 กำหนดสีของพื้นและข้อความในเมนูแต่ละชุด ถ้าเปลี่ยนค่าในบรรทัดที่ 2 ก็ต้องเพิ่มค่าในกลุ่มนี้ด้วย ถ้าใช้กับจอวีจีเอโมโนโครมอาจต้องกำหนดสีบางสีใหม่เพื่อความเหมาะสม

3. บรรทัดที่ 8-9 กำหนดข้อความแสดงชื่อของแต่ละเมนู

4. บรรทัดที่ 10 ตัวแปรรองรับจำนวนรายการสูงสุดแต่ละเมนู

```
 1: uses crt,dos,graph,thai,Head;
 2: const Max=3;
 3:       CPatt0:array[1..Max] of byte=(10,11,14);
 4:       CPatt1:array[1..Max] of byte=(12,9,13);
 5:       CPatt2:array[1..Max] of byte=(4,1,5);
 6:       ColorText:array[1..Max] of byte=(14,15,11);
 7:       CMenu:array[1..Max] of byte=(1,12,6);
 8:     HeadMenu:array[1..Max] of string=
 9:       ('การจัดการไฟล์ข้อมูล','  งานด้านปกครอง','  งานด้านแนะแนว');
10: var MaxMenu:array[1..Max] of byte;
11:     Choice,Column,k:byte;
12:     MenuST:array[1..Max,1..8] of string;
13:     XMenu,Ymenu:array[1..Max] of word;

14: PROCEDURE ShowMenu(No:byte);
15: var k,n,p:byte; ST:string;
16: begin
17:   Board3D(Xmenu[No]-8,Ymenu[No]-42,Xmenu[No]+MaxColumn*9+27,
18:                       Ymenu[No]+MaxMenu[No]*30+20,Cmenu[No],0);
19:   Board3D(Xmenu[No],Ymenu[No]-35,Xmenu[No]+MaxColumn*9+20,
20:                                  Ymenu[No]-5,Cpatt1[No],1);
21:   setcolor(ColorText[No]);
22:   Outthaixy(Xmenu[No]+20,Ymenu[No]-32,#17+HeadMenu[No]);
23:   setviewport(Xmenu[No],YMenu[No],getmaxx,getmaxy,clipon);
24:   Board3D(0,0,MaxColumn*9+20,MaxMenu[no]*30+13,Cpatt0[no],1);
25:   for k:=1 to MaxMenu[no] do
26:    begin
27:     ST:=''; N:=0;
28:     for p:=1 to length(MenuST[no][k]) do
29:      begin
30:        if MenuST[no][k][p] in [#32..#214] then Inc(n);
31:        if N<=MaxColumn-2 then ST:=ST+MenuST[no][k][p];
32:      end;
33:     MenuST[no][k]:=ST;
34:     Block3D(10,k*30-22,MenuST[no][k],CPatt1[no],ColorText[No],0);
35:    end;
36:   setviewport(0,0,getmaxx,getmaxy,clipon);
37: end;
```

รูปที่ 6
การกำหนดค่าคงที่ ตัวแปรใช้งาน และการแสดงรายการในเมนู

5. บรรทัดที่ 12 กำหนดตัวแปรรองรับรายการของเมนูแต่ละชุด

6. บรรทัดที่ 13 ตัวแปรรองรับมุมบนซ้ายของแต่ละเมนู

7. บรรทัดที่ 14-37 ส่วนการแสดงรายการในเมนูชุดที่กำหนด (ตัวแปร No)

8. บรรทัดที่ 17-18 แสดงกระดาน 3 มิติแบบนูน แสดงขอบเขตของรายการทั้งหมดในเมนูนั้น ๆ (รวมข้อความส่วนหัวของเมนูด้วย)

9. บรรทัดที่ 19-22 แสดงข้อความส่วนหัวของเมนูบนกระดาน 3 มิติแบบเว้า

10. บรรทัดที่ 23 กำหนดขอบเขตของจอภาพตามค่ามุมบนซ้ายของเมนูแต่ละชุด ซึ่งช่วยให้เรากำหนดตำแหน่งของเมนูชุดใดอยู่ตรงไหนของจอภาพก็ได้

11. บรรทัดที่ 27-33 ส่วนการตรวจสอบความกว้างของข้อความในเมนูแต่ละรายการ ให้อยู่ในขอบเขตของตัวแปร Maxcolumn-2 ซึ่งต่างจากแบบแรกที่จะให้กว้างเท่าใดก็ได้ รายการใดมีจำนวนคอลัมน์มากก็จะตัดข้อความส่วนท้าย ๆ ทิ้งไป พร้อมกับกำหนดให้ข้อความของรายการนั้นเท่ากับข้อความที่คำนวณได้ด้วย

12. บรรทัดที่ 34 แสดงรายการของเมนูลงบนแป้น 3 มิติ

13. บรรทัดที่ 36 จบลงด้วยการคืนขอบเขตของจอภาพให้เป็นปกติ

การเลือกรายการที่ยากขึ้น

เนื่องจากเราต้องการเขียนชุดคำสั่งที่เป็นมาตรฐานให้มากที่สุดเท่าที่จะมากได้ ซึ่งตามที่ผู้เขียนกำหนดไว้แล้วว่าจะต้องเป็นโปรแกรมที่ปรับเปลี่ยนค่าที่เกี่ยวข้องได้โดยง่าย อีกทั้งต้องสามารถเปลี่ยนตำแหน่งของแต่ละเมนูได้อย่างมีประสิทธิภาพ กล่าวคือ ไม่ใช่วางเมนูเรียงลำดับจากซ้ายไปขวาเป็นเมนูที่ 1,2,3 ตามลำดับ จึงทำให้ต้องใช้ความคิดในการพัฒนาโปรแกรมมากขึ้น

แนวคิดที่ทำได้ก็คือ กำหนดค่าตัวแปรให้มุมบนซ้ายของแต่ละเมนู เมื่อมีการเลือกรายการในเมนูด้วยการกดปุ่มของเม้าส์ เราก็คำนวณหาว่าขณะนั้นเม้าส์กำลังอยู่ในเมนูชุดใด และอยู่ตำแหน่งรายการลำดับที่เท่าใดด้วย

สิ่งที่เราต้องมาใช้ตรวจสอบคือการหาตำแหน่งของเม้าส์จากตัวแปร XMouse, YMouse เทียบกับตัวแปรที่เป็นมุมซ้ายของเมนู โดยต้องทราบความกว้างและความสูงของรายการทั้งหมดในเมนูชุดนั้นด้วย

ขอให้ผู้อ่านลองใช้ความคิดด้วยตนเองก่อนที่จะไปดูโปรแกรมในรูปที่ 7 โดยรายละเอียดของโปรแกรมในรูปที่ 7 มีดังนี้

1. บรรทัดที่ 39-40 ส่วนกำหนดไว้ตรวจสอบว่าเป็นรายการลำดับที่เท่าใด

2. บรรทัดที่ 42-43 ชุดคำสั่งจบการทำงานของโปรแกรมและกลับไปยังดอส

3. บรรทัดที่ 44-56 ชุดคำสั่งตรวจสอบลำดับที่ของรายการที่เลือกโดยลักษณะการทำงานคล้ายกับที่เขียนในไฟล์ menu__no1.pas โดยเพิ่มการกำหนดว่าเป็นเมนูใด (ตัวแปร choice) เท่านั้น

4. บรรทัดที่ 57-68 ชุดคำสั่งตรวจสอบว่าอยู่ในเมนูใด โดยถ้ากดปุ่มเม้าส์ที่นอกรายการเมนูทั้งหมด จะได้ตัวแปร column = 0 การตรวจสอบนั้นเราจะตรวจขอบเขตของมุมบนซ้ายและ

รูปที่ 7
การเลือกรายการในเมนูแบบที่มีรายการมากชุด

```
38: PROCEDURE SlideMenu;
39: const Ypix:array[0..10] of word=
40:          (8,38,68,98,128,158,188,218,248,278,308);
41: var key:boolean; ColumnOld,k:byte;
42:       Procedure Stop;
43:        begin closegraph; halt; end;
44:       Procedure Checkchoice;
45:       begin
46:         choice:=0;
47:         if Ymouse>Ypix[0] then
48:          begin
49:            Repeat
50:              inc(choice);
51:            until Ypix[choice]>Ymouse;
52:          end;
53:        if Choice in [1..MaxMenu[Column]] then
54:         Block3D(10+XMenu[Column],choice*30-22+Ymenu[Column],
55:           MenuST[Column][Choice],Cpatt2[Column],ColorText[Column],1),
56:       end;
57:       Procedure CheckColumn;
58:       begin
59:          Column:=0; k:=1;
60:          repeat
61:           if (XMouse>Xmenu[k]+11) and
62:              (XMouse<Xmenu[k]+MaxColumn*9+6) and
63:              (Ymouse>8+YMenu[k]) and
64:              (Ymouse<Ypix[MaxMenu[k]]-5+YMenu[k]) then
65:                begin Column:=k; k:=Max;  end;
66:           inc(k);
67:          until k>Max;
68:       end;
```

รูปที่ 7 (ต่อ)

```
69: begin
70:   choice:=1; Column:=1; key:=false; ColumnOld:=1;
71:   repeat
72:    repeat
73:      if Mouse then
74:        begin
75:         Mouseon; regs.AX:=3; Intr($33,regs);
76:         if Regs.BX in [1..2] then
77:          begin
78:           MouseOff;
79:           if (Column<>0) and key then
80:            Block3D(10+XMenu[Column],choice*30-22+YMenu[Column],
81:              MenuST[Column][Choice],Cpatt1[Column],ColorText[Column],0);
82:           Findxy(XMouse,YMouse);
83:           if (Xmouse>530) and (Xmouse<630) and (Ymouse>320)
84:                          and (YMouse<340) then Stop;
85:           CheckColumn;
86:           if Column in [1..Max]  then
87:            begin
88:             Dec(YMouse,YMenu[Column]);
89:             if ((YMouse-8) mod 30 in [2..24]) and
90:                 (Ymouse>8) and
91:                 (Ymouse<Ypix[MaxMenu[Column]]-5) then
92:                  begin
93:                   Checkchoice;  ColumnOld:=Column;
94:                   if choice in [1..MaxMenu[Column]] then
95:                       begin   MouseOff; Exit;  end;
96:                  end;
97:            end
98:              else Column:=ColumnOld;
99:          end;
100:       end;
101:    if keypressed then
102:     begin
103:       readchar;
104:       if Mouse then MouseOff;
105:       if (ch=#13) and key then exit;
106:       if ch=#27 then Stop;
107:     end;
108:  until Funckey in [#72,#75,#77,#80];
109:   Block3D(10+XMenu[Column],choice*30-22+YMenu[Column],
110:      MenuST[Column][Choice],Cpatt1[Column],ColorText[Column],0);
111:   case funckey of
112:    #72:if choice>1 then Dec(Choice) Else Choice:=MaxMenu[Column];
113:    #80:if choice<MaxMenu[Column] then Inc(Choice) Else Choice:=1;
114:    #75:begin
115:         if column>1 then Dec(Column) Else Column:=Max;
116:         if choice>MaxMenu[Column] then choice:=MaxMenu[Column];
117:        end;
118:    #77:begin
119:         if column<Max then Inc(Column) Else Column:=1;
120:         if choice>MaxMenu[Column] then choice:=MaxMenu[Column];
121:        end;
122:   end;
123:    Block3D(10+XMenu[Column],choice*30-22+YMenu[Column],
124:        MenuST[Column][Choice],Cpatt2[Column],ColorText[Column],1);
125:    key:=true; Funckey:=#0;
126:    setposition(Xmenu[Column]+100,Ymenu[Column]-5+Choice*30);
127:  until ch=#255;
128: end;
```

มุมล่างขวาของข้อความในเมนูชุดนั้น ๆ (รูปสี่เหลี่ยมที่ล้อมข้อความของเมนู) ซึ่งทำให้ได้ตัวแปร column เป็นลำดับที่ของเมนู

5. บรรทัดที่ 70 กำหนดตัวแปรเริ่มต้น

6. บรรทัดที่ 75 ตรวจการเคลื่อนที่ของเม้าส์

7. บรรทัดที่ 76-99 ส่วนการทำงานเมื่อกดปุ่มใดปุ่มหนึ่งของเม้าส์

8. บรรทัดที่ 78-81 แสดงรายการในเมนูที่ใช้แป้นลูกศรเลื่อนไปก่อนหน้าให้คืนสภาพปกติ

9. บรรทัดที่ 82 หาตำแหน่งของเม้าส์

10. บรรทัดที่ 83-84 ถ้ากดปุ่มของเม้าส์ที่ตำแหน่งบนจอภาพจะมีข้อความว่า Esc เลิกงาน (มีคำสั่งให้แสดงบนแป้น 3 มิติแบบเว้าในโปรแกรมรูปที่ 8) จะเลิกงานและกลับไปยังดอส

11. บรรทัดที่ 85 ถ้าไม่อยู่ในเงื่อนไขของบรรทัดที่ 83-84 ก็จะทำการตรวจว่าอยู่ในเมนูใด

12. บรรทัดที่ 86-98 ส่วนการตรวจสอบว่าอยู่ในรายการลำดับที่เท่าใดของเมนูนั้น ๆ ให้สังเกตคำสั่งตรวจสอบหาค่า YMouse จะต้องอยู่ในเงื่อนไขของบรรทัดที่ 89-91 ก่อน จึงจะทำการหาค่าของ Choice ถ้าตรวจพบว่าการกดปุ่มของเม้าส์อยู่ในตำแหน่งรายการในเมนูนั้น ๆ ก็จะออกจากชุดคำสั่ง Slide-menu โดยจะได้ค่า Choice นำไปใช้งาน

13. บรรทัดที่ 101-126 ส่วนการทำงานเมื่อกดแป้นลูกศรขึ้นลงหรือลูกศรซ้ายขวา ซึ่งคิดว่าผู้อ่านคงทำความเข้าใจด้วยตนเองได้ จึงของดอธิบาย

โปรดสังเกตเงื่อนไขการจบการวนรอบ ซึ่งไม่มีโอกาสเป็นจริง เพราะว่าการออกจากชุดคำสั่งนี้ในกรณีของการใช้แป้นพิมพ์คือ ให้เลือกรายการด้วยแป้นลูกศรและสิ้นสุดการตัดสินใจเลือกด้วยการกดแป้น Enter (ตามบรรทัดที่ 105) และเลิกงานด้วยการกดแป้น Esc เท่านั้น

คราวนี้ก็มาถึงส่วนการแสดงผลหน้าจอภาพของแต่ละเมนูและข้อความอื่น ๆ ที่เกี่ยวข้อง ซึ่งมีเทคนิคที่ช่วยให้เราจัดรูปแบบหน้าจอได้ตามความต้องการ โดยเพียงแต่เปลี่ยนตัวเลขเท่านั้น ในกรณีตัวอย่างต่อไปนี้ผู้เขียนไม่ได้กำหนดเมนูแบบที่เห็นกันทั่ว ๆ ไป คือ จากซ้ายไปขวา รายการเลือกในเมนูเรียงลำดับแถวให้รายการแรกอยู่ในระดับเดียวกัน และแยกแต่ละเมนูออกจากกันอย่างชัดเจน โดยผู้เขียนขอกำหนดค่าตัวแปรที่ทำให้เมนูมีการซ้อนทับกันในบางส่วน และไม่ได้เรียงลำดับจากซ้ายไปขวา

โดยโปรแกรมในรูปที่ 8 เป็นตัวอย่างหนึ่งของการแสดงรายการแบบมีเมนูมากชุดให้ปรากฏบนจอภาพ ขอให้ผู้อ่านลองพิจารณาคำสั่งประกอบกับผลของการรันไฟล์นี้ในรูปที่ 9 ว่าท่านมีความเข้าใจการทำงานต่าง ๆ หรือไม่

สำหรับโปรแกรมในรูปที่ 8 นี้ ผู้เขียนขออธิบายแบบคร่าว ๆ ดังนี้

1. บรรทัดที่ 132 ทุกครั้งก่อนจะเขียนภาพหรือข้อความใดจะต้องการให้หยุดการแสดงเคอร์เซอร์ของเม้าส์ก่อนเสมอ

2. บรรทัดที่ 134-135 วาดภาพกระดาน 3 มิติและแสดงข้อความให้ทราบว่าเป็นเรื่องใด

3. บรรทัดที่ 136-138 รายการเลือกในเมนูชุดที่ 1 มี 6 รายการ

4. บรรทัดที่ 139-140 รายการเลือกในเมนูชุดที่ 2 มี 3 รายการ

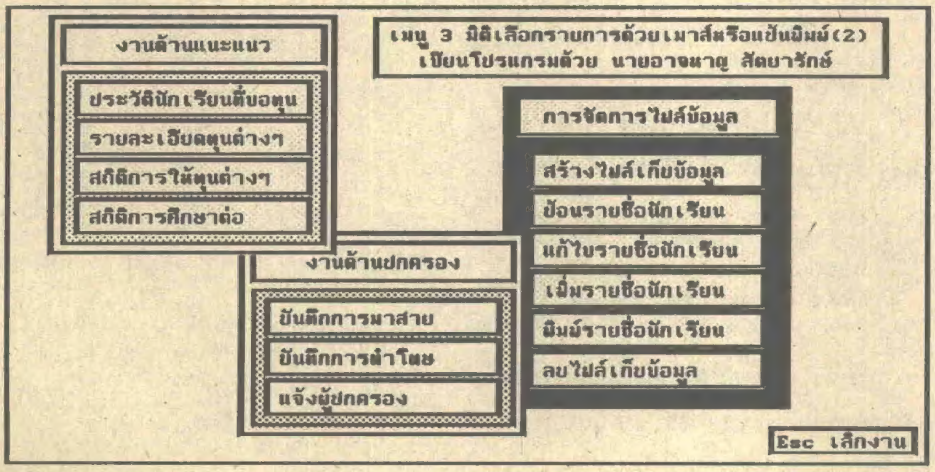
รูปที่ 8
ส่วนของโปรแกรมหลักที่กำหนดหน้าจอของเมนูที่มีรายการมากชุด

```
129: PROCEDURE MainMenu;
130: begin
131:  Repeat
132:   if Mouse then MouseOff;          MaxColumn:=18;
133:   Board3D(0,0,639,getmaxy,3,0);  Board3D(255,10,609,55,6,1);
134:   OutThaixy(265,10,#17#3'6'#1'เมนู 3 มิติเลือกรายการด้วยเมาส์หรือแป้นพิมพ์(2)');
135:   OutThaixy(285,30,#17'เขียนโปรแกรมด้วย นายอาจหาญ สัตยารักษ์');
136:   MenuST[1][1]:='สร้างไฟล์เก็บข้อมูล';    MenuST[1][2]:='ป้อนรายชื่อนักเรียน';
137:   MenuST[1][3]:='แก้ไขรายชื่อนักเรียน';  MenuST[1][4]:='เพิ่มรายชื่อนักเรียน';
138:   MenuST[1][5]:='พิมพ์รายชื่อนักเรียน';  MenuST[1][6]:='ลบไฟล์เก็บข้อมูล';
139:   MenuST[2][1]:='บันทึกการมาสาย';        MenuST[2][2]:='บันทึกการทำโทษ';
140:   MenuST[2][3]:='แจ้งผู้ปกครอง';
141:   MenuST[3][1]:='ประวัตินักเรียนที่ขอทุน'; MenuST[3][2]:='รายละเอียดทุนต่างๆ';
142:   MenuST[3][3]:='สถิติการให้ทุนต่างๆ';    MenuST[3][4]:='สถิติการศึกษาต่อ';
143:   MaxMenu[1]:=6;   MaxMenu[2]:=3;   MaxMenu[3]:=4;
```

รูปที่ 8 (ต่อ)

```
144:   XMenu[1]:=353;    YMenu[1]:=105;
145:   XMenu[2]:=170;    YMenu[2]:=216;
146:   XMenu[3]:=38;     YMenu[3]:=50;
147:   for k:=1 to Max do ShowMenu(k);
148:   Board3D(530,320,630,340,7,1);
149:   OutThaixy(535,318,#17#3'3'#1'Esc เลิกงาน');
150:   WindowMouse(0,0,getmaxx-8,getmaxy-8);
151:   setposition(Xmenu[1]+100,Ymenu[1]+25);
152:   SlideMenu;
153:   block3D(Xmenu[Column]+10,Ymenu[Column]+
154:        Maxmenu[Column]*30+25,MenuST[Column][choice],9,15,0);
155:   readmouse;
156:   case Column of
157:    1:case choice of
158:       1..10:{procedure for run}
159:      end;
160:    2:case choice of
161:       1..10:{procedure for run}
162:      end;
163:    3:case choice of
164:       1..10:{procedure for run}
165:      end;
166:   end;
167:  until ch=#255;
168: end;

169: {=================== MENU_NO2.PAS ====================}
170: begin
171:   Mainmenu;
172:   closegraph;
173: end.
```



รูปที่ 9 ภาพหน้าจอจากการรันไฟล์ menu_no.2.pas

5. บรรทัดที่ 141-142 รายการเลือกในเมนูชุดที่ 3 มี 4 รายการ

6. บรรทัดที่ 143 กำหนดจำนวนสูงสุดของเมนูแต่ละชุด

7. บรรทัดที่ 144-146 กำหนดตำแหน่งมุมบนซ้ายของเมนูแต่ละชุด

8. บรรทัดที่ 147 แสดงเมนูแต่ละชุดบนจอภาพ คำสั่งนี้ผู้อ่านอาจแยกเขียนเป็นของแต่ละชุดก็ได้ ทั้งนี้ขึ้นอยู่กับความต้องการการแสดงผลบนจอภาพ (ประกอบกับการให้ค่าตัวแปรในบรรทัดที่ 144-146 ใหม่) เช่น ต้องการให้เมนู 1 อยู่ตรงกลาง โดยซ้อนอยู่บนเมนู 2 (ทางขวา) และเมนู 3 (ทางซ้าย) ก็อาจกำหนดได้ดังนี้

```
XMenu[1]:=227;  YMenu[1]:=130;
XMenu[2]:=410;  YMenu[2]:=170;
XMenu[3]:=40;   YMenu[3]:=70;
ShowMenu(2); ShowMenu(3); Show-
Menu(1);
```

แต่ถ้าเปลี่ยนเป็น

```
ShowMenu(2); ShowMenu(1); Show-
Menu(3);
```

ผลบนจอภาพจะเป็นการซ้อนทับของเมนู 2, 1 และ 3 เรียงจากขวาไปซ้าย ขอให้เปลี่ยนค่าแล้วรันดูการแสดงผลบนจอภาพเอง

9. บรรทัดที่ 148-149 แสดงข้อความให้ทราบว่าการเลิกงานทำอย่างไร

10. บรรทัดที่ 150-152 ส่วนการทำงานเพื่อเลือกรายการในเมนู

11. บรรทัดที่ 153-155 ส่วนนี้เป็นเพียงต้องการให้ผู้อ่านตรวจสอบว่าการทำงานของโปรแกรมทั้งหมดถูกต้องหรือไม่เท่านั้น เมื่อใช้งานจริงให้ตัดทั้งสามบรรทัดนี้ออกไป

ส่วนที่เหลือของโปรแกรมคงทำความเข้าใจได้

ขอให้ผู้อ่านลองเปลี่ยนคำสั่งในบรรทัดที่ 144-147 ใหม่ด้วยตนเองแล้วรันดูผลการทำงาน ซึ่งจะช่วยให้เข้าใจว่าแท้จริงแล้วการจัดรูปแบบหน้าจอนั้นเป็นเรื่องง่าย ๆ

โปรแกรมในรูปที่ 6-8 คือไฟล์ menu__no2.pas

เพื่อให้ผู้อ่านมีความรู้ในเรื่องหนึ่งซึ่งอาจยังไม่มีใครเขียนถึงเช่นกัน ความรู้นั้นคือยูนิต thai.tpu ซึ่งผู้เขียนได้กำหนดให้เข้ากราฟิกขนาด 640×350 จุดไว้ เพราะว่าเป็นโหมดที่มีกราฟิก 2 หน้า ซึ่งช่วยให้เรานำมาใช้ประโยชน์ได้ แต่บนจอสีหรือจอวีจีเอโมโนโครมนั้นมีกราฟิกแบบอื่น ๆ อีกคือ ขนาด 640×480 จุด (หนึ่งหน้ากราฟิก) และขนาด 640×200 จุด (4 หน้ากราฟิก)

ปัญหาจึงมีว่า...ถ้าเราต้องการให้เป็นกราฟิก 640×480 จุดเราจะทำอย่างไร ต้องเข้าไปแก้ไขยูนิต thai.tpu ใหม่หรือไม่ คำตอบของผู้เขียนคือไม่ต้องแก้ไข เพราะว่าเทอร์โบปาสคาลได้ให้คำสั่งช่วยงานมาแล้วคือคำสั่ง SetGraphMode (Mode:integer);

- ต้องการกราฟิกขนาด 640×200 จุด ให้กำหนดด้วย SetGraphMode(0);
- ต้องการกราฟิกขนาด 640×350 จุด ให้กำหนดด้วย SetGraphMode(1);
- ต้องการกราฟิกขนาด 640×480 จุดให้กำหนดด้วย SetGraphMode(2);

โดยไม่คำนึงว่าตอนเข้ามายังกราฟิกโหมดนั้นเป็นกราฟิกขนาดใด

ที่น่าสนใจมากที่สุดของคำสั่งนี้อยู่ที่ เราสามารถกำหนดขนาดกราฟิกสลับกันในโปรแกรมหนึ่ง ๆ ได้ เช่น

ในเมนูเลือกที่มีรายการมากก็ใช้ SetGraphMode(2); เมื่อเลือกให้รันเนื้อเรื่องอื่น ๆ อาจกำหนดให้เป็น SetGraphMode(1); หรือ SetGraphMode (0); ตามต้องการได้

ถ้าผู้อ่านกำลังรันบนจอสีหรือจอวีจีเอโมโนโครมให้ลองเพิ่มคำสั่งต่อไปนี้ดู

1. เพิ่มคำสั่ง SetGraphMode(2); แทรกก่อนบรรทัดที่ 132

2. เปลี่ยนคำสั่งในบรรทัดที่ 153-154 มาเป็นคำสั่งกลุ่มนี้แทน

```
setgraphmode (Column mod 3);
block3D(Xmenu[Column]+10,120,
MenuST[Column][choice],9,15,0);
```

เมื่อเรียบร้อยแล้วให้รันดูผลบนจอภาพ เมื่อเลือกรายการในเมนูแต่ละชุดให้สังเกตดูขนาดของแป้นและข้อความบนจอภาพที่มีขนาดต่างกันตามกราฟิกที่แสดงผลอยู่

ในกรณีที่รันบนกราฟิกขนาด 640×480 จุด คงต้องเปลี่ยนคำสั่งในบรรทัดที่ 148, 149, 83 และ 84 ใหม่ เพื่อให้ตำแหน่งของข้อความว่า Esc เลิกงานเลื่อนลงมาอยู่ที่ข้างล่างของจอภาพ

บทความเรื่องนี้จึงให้ความคิดในการเขียนกราฟิกใหม่หลายประการ ซึ่งคงจะเป็นประโยชน์ต่อผู้อ่านในการพัฒนาโปรแกรมคราวต่อไปให้มีรูปแบบที่แปลกตายิ่งขึ้น

ส่งท้าย

สำหรับบทความในวารสารฯฉบับต่อจากนี้เป็นต้นไป ผู้เขียนจะขอให้ยูนิต thai.tpu, head.tpu ที่ได้กล่าวถึงในฉบับนี้เป็นแบบในการเขียนโปรแกรมทั้งหมด ดังนั้นถ้าผู้อ่านสนใจในบทความของผู้เขียนก็ควรจะเก็บทั้งสองยูนิตนี้ไว้ หากมีปัญหาเรื่องยูนิต thai.tpu ก็ขอให้ติดต่อใช้บริการซอฟต์เมล์จากทางวารสารฯได้

สำหรับฉบับนี้คงต้องบอกว่า ลาก่อน...